

Software Assurance Considerations for STANAG 4671

Kelly Hayhurst

NASA Langley Research Center

Hampton, Virginia, USA

kelly.j.hayhurst@nasa.gov

NATO UAS Safety Objectives Workshop

14 September 2010



Langley Research Center

Outline

- **Software assurance in the STANAG 4671 context**
- **Fundamental concepts from RTCA/DO-178B**
- **Factors to consider**
 - differences in software assurance guidance for airborne and ground-based systems
 - differences between software assurance levels
 - changes proposed for DO-178B & DO-278
- **Concluding thoughts**



Software Assurance

What is it? Planned and systematic actions necessary to provide adequate confidence and evidence that a software product satisfies given requirements

- **For AMC.1309, actions should comply with the RTCA/DO-178B "Software Considerations in Airborne Systems and Equipment Certification"**
 - non-airborne, non-flight critical software should comply with RTCA/DO-278 "Guidelines for Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems Software Integrity Assurance"
- **The extent of the actions required is specified by software level, as identified through system safety assessment**

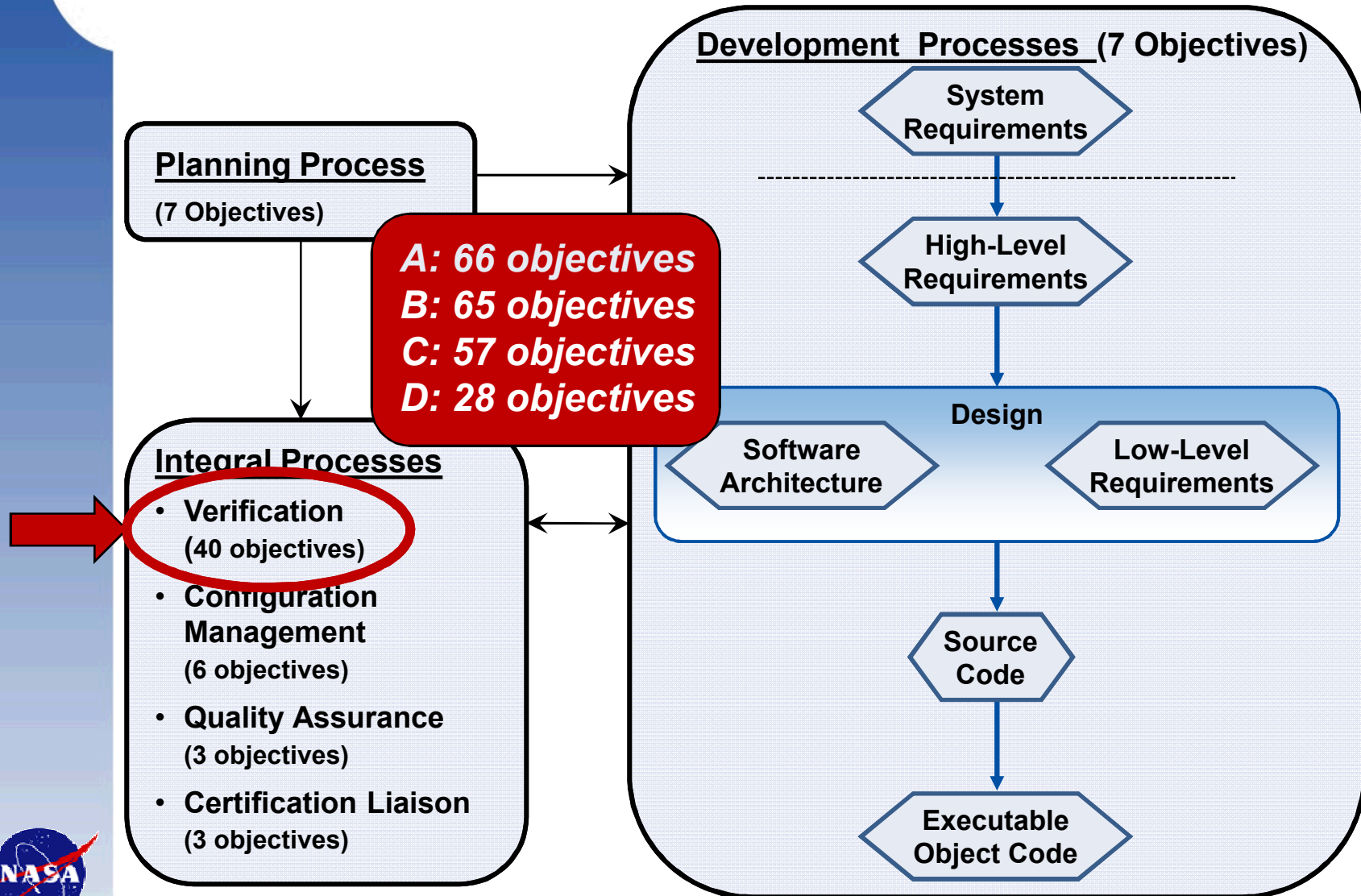


Software Levels for AMC.1309

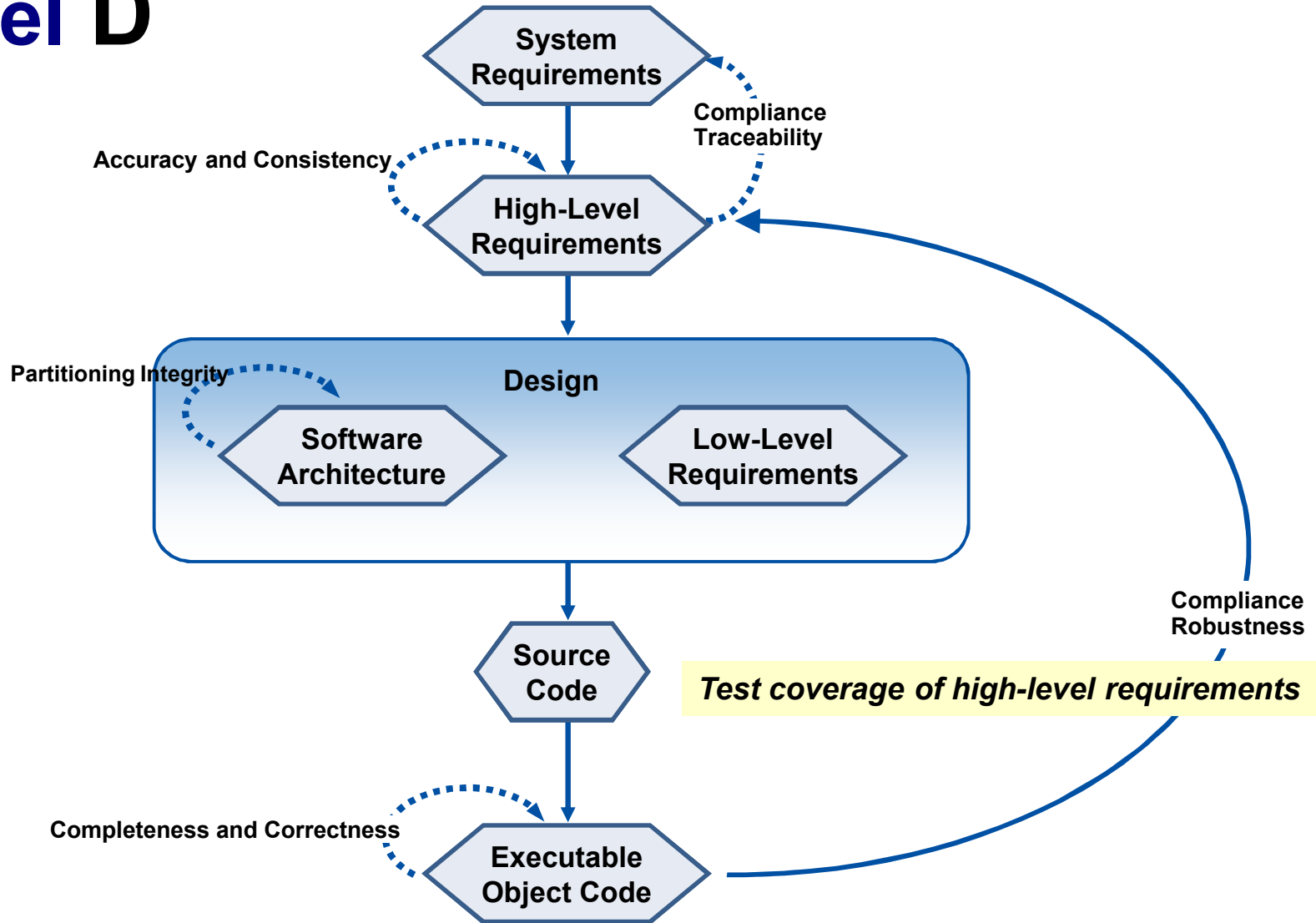
DAL allocation for system and each portion of the system architecture		Degree of redundancy	
		Single failure/errors	Double failure/errors two or more independent/dissimilar portions
Failure Conditions Classification	Catastrophic	DAL B	DAL B for the system and, DAL C for each portion
	Hazardous	DAL C	DAL C for the system and, DAL D for each portion
	Major	DAL D	DAL D for the system and, DAL D for each portion
	Minor	DAL E	DAL E
	No Safety Effect	DAL E	



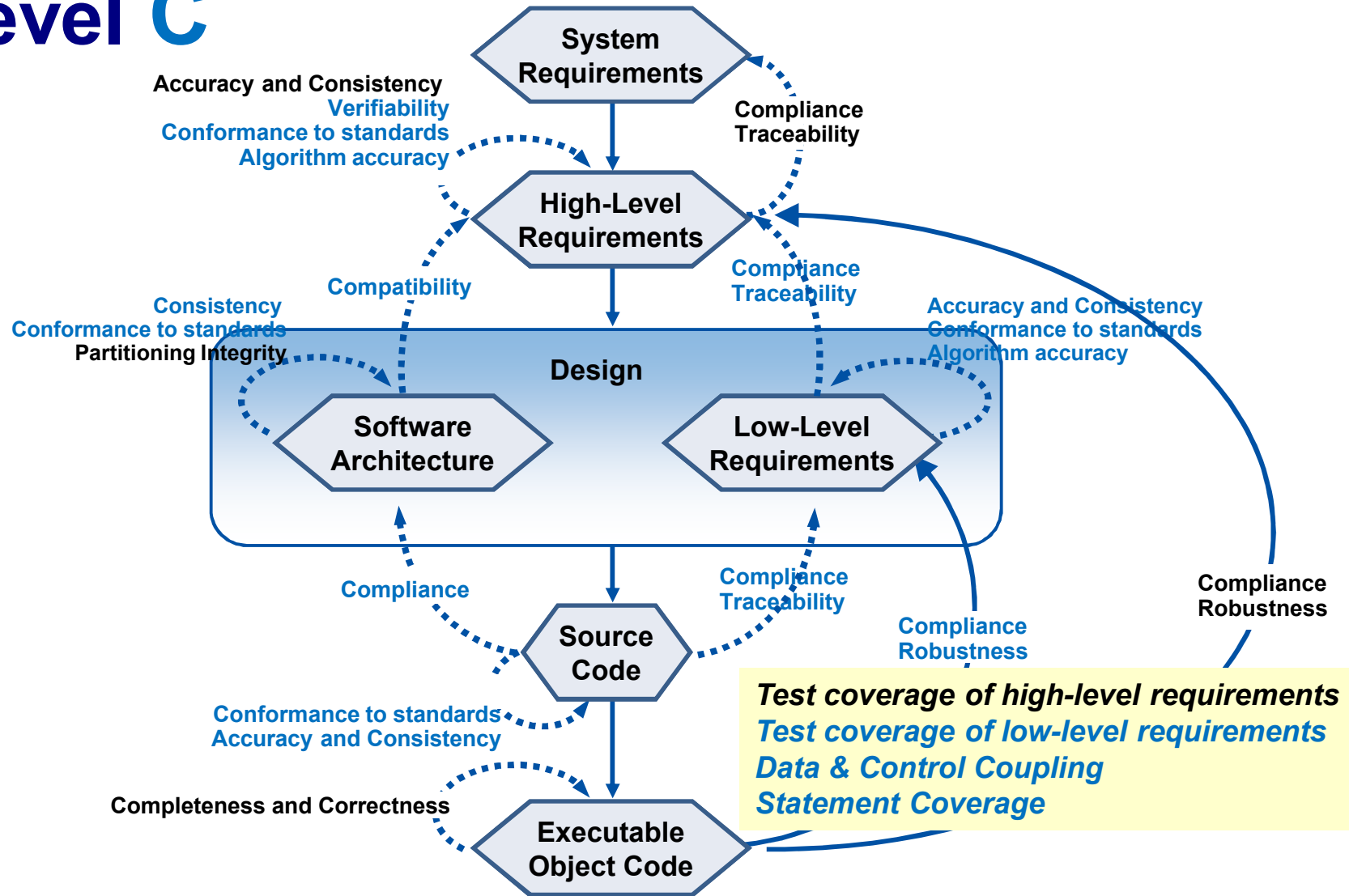
DO-178B Life Cycle Processes



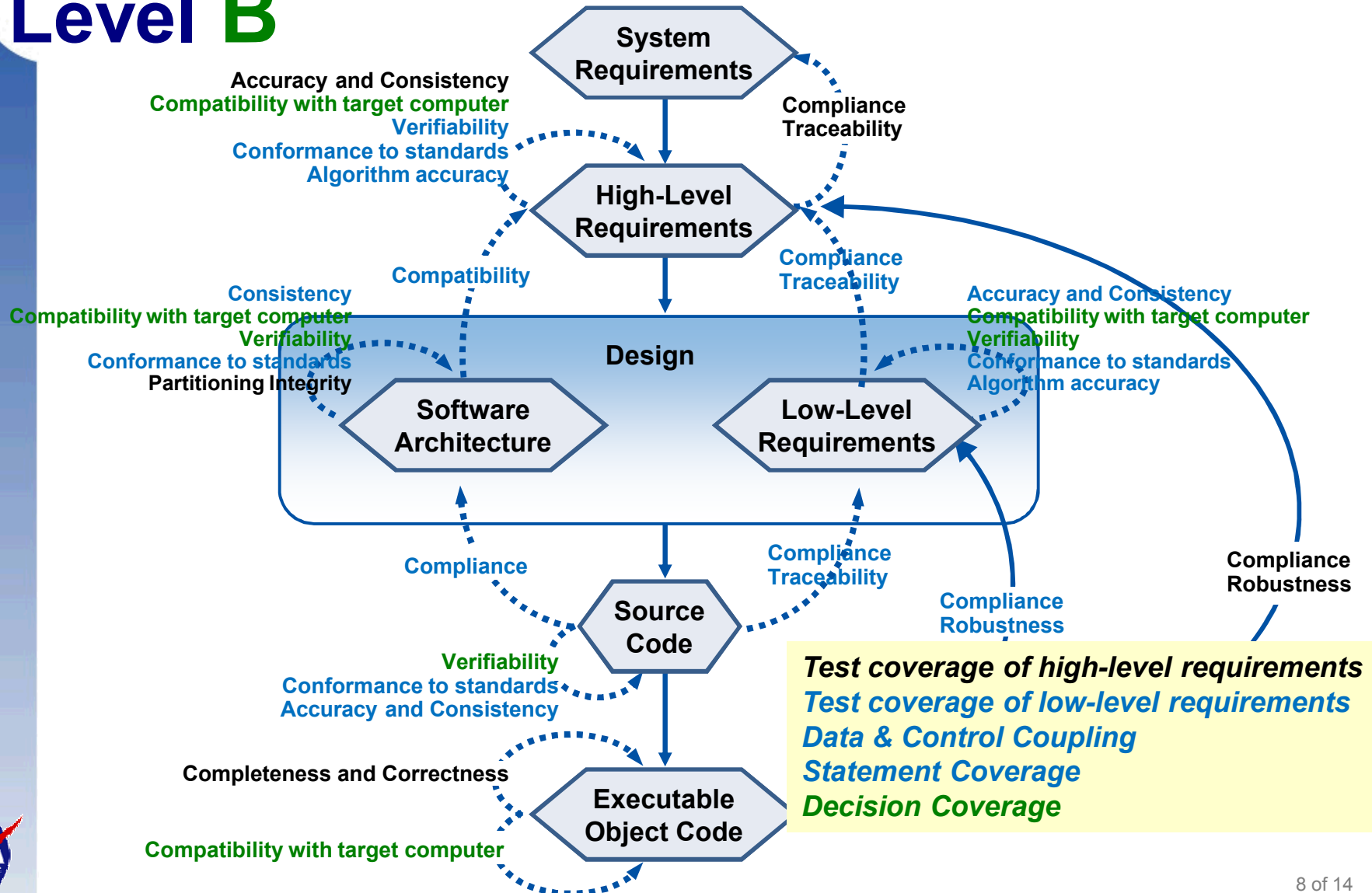
DO-178B Verification Objectives for Level D



DO-178B Verification Objectives for Level C



DO-178B Verification Objectives for Level B



Differences in Guidance

DO-178B

- **5 Software Levels** →
 - A-E
- **Development** →
 - high & low level requirements, architecture, source and executable code required at all levels
- **Planning/CM/QA/Cert Coordination** →
- **Verification** →

DO-278

- **6 Assurance Levels**
 - AL3/C: *Hazardous in AMC.1309*
+ AL4
 - AL5/D: *Major in AMC.1309*
- **Same objectives**
 - low level requirements and source code **not** required for AL4 & AL5
 - architecture **not** required for AL5
 - + 1 objective for Adaptation Data
- **Same objectives**
 - + assurance of transition criteria for AL3
- **Same objectives**
 - + architecture compatibility with target
- + **Objectives for Commercial Off-the-Shelf software**



Considerations for Airborne versus Non-airborne Software

- **Differences between DO-178B & DO-278 are not huge**
 - but they are not insignificant
- **Joint committee RTCA SC-205/EUROCAE WG-71 recommended a single software guidance document for airborne and ground-based software**
 - RTCA rejected that recommendation

Should the objectives for software assurance depend on where functions reside in a UAV system?



Considerations for Software Design Assurance Levels

Severity	Key factors	Level
Catastrophic	• uncontrolled flight or uncontrolled crash, which can potentially result in a fatality • potential fatality to UAV crew or ground staff	B
Hazardous	• controlled-trajectory termination or forced landing, with potential loss of the UAV • no fatality likely, but potential for serious injury to UAV crew or ground staff	C
Major	• emergency landing of the UAV on a predefined site • no serious injury likely, but potential for injury to UAV crew or ground staff	D
Minor	• slight reduction in safety margins or functional capabilities, and slight workload increase	E

Do incidents/accidents/problems attributed to software anomalies support these levels?

e.g., are the rates of occurrence of catastrophic events extremely improbable ($<10^{-6}$ per flight hour)?

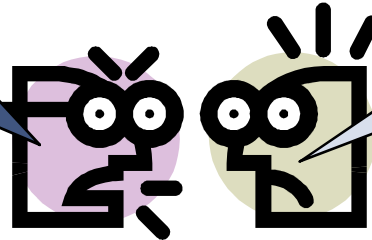
Is there sufficient data to know?



Level A, if more is needed

- The primary difference between Level A & B is the modified condition/decision coverage (MC/DC) objective
- The intent of the MC/DC objective is to:
 - increase confidence in requirements correctness
 - increase confidence that testing was sufficient
 - increase confidence that there is no dead or unintended code
- The value of MC/DC is often debated and misunderstood

It costs a fortune
– and doesn't find
a single bug!



That's right! Planning
for MC/DC helps us
catch problems early.

Is there a need to consider Level A or something else?

Pending Updates* to Software Standards

- **DO-178B & DO-278 are being updated to**
 - provide clear and consistent ties with the systems and safety processes
 - address emerging software trends and technologies
 - implement an approach that can change with technology
- **Proposed changes to core text are relatively minor**
 - changes to description of system aspects related to software
 - 5 new objectives (1 for planning, 2 for verification of parameter data, and 2 for quality assurance)
- **New proposed "technology supplements" for**
 - Object Oriented Technology, Model-based Development, and Formal Methods
 - Tool Qualification guidance is also being updated



Could these changes impact STANAG 4671?

(*RTCA SC-205 and EUROCAE WG-71 are still working on these updates. Proposed changes and proposed new documents have not gone through any public review and comment yet, and should not be considered approved/final in any way.)



Concluding Thoughts

- **Software assurance plays an essential role in meeting STANAG 4671 safety objectives**
 - in making sure that software does what it is supposed to do and doesn't do what it is not supposed to do
 - *but determining those requirements for UAVs is not easy*
- **Understanding software problems, especially those occurring in operations, could help a lot**
 - to confirm or dispute the effectiveness of assurance requirements
 - *but details about software problems are often unavailable*
- **Sharing experiences and data is crucial to developing the best possible assurance practices and policies**

